**DEVOPS**

**Assignment – 1**

SANJAY KRISHNA M

4NI19IS083

7th semester, ‘A’ section

1. **What is CI / CD?**

CI and CD stand for continuous integration and continuous delivery/continuous deployment. In very simple terms, CI is a modern software development practice in which incremental code changes are made frequently and reliably. Automated build-and-test steps triggered by CI ensure that code changes being merged into the repository are reliable. The code is then delivered quickly and seamlessly as a part of the CD process. In the software world, the CI/CD pipeline refers to the automation that enables incremental code changes from developers’ desktops to be delivered quickly and reliably to production.

Continuous integration is when development teams implement small code changes and continuously merge their work to a shared repository. With continuous integration in place, code changes happen at a more frequent pace. The coding work is broken down into smaller parts, making it easier to spot bugs and issues before releasing to customers.

After CI, we move to continuous delivery. It's the natural next step after CI, as the changes made during CI need to be delivered to the customers or a sub-set of them. This is the staging and testing environment where the code is automatically reviewed for deployment readiness through tests, and where issues are flagged.

1. **What are feature flags?**

Feature flags is a software engineering technique that turns select functionality on and off during runtime, without deploying new code. A feature flag is the ability to turn features of your application on/off at easily. A common use case for feature flags is to enable certain features on a per-user basis. This enables teams to make changes without pushing additional code and allows for more controlled experimentation over the lifecycle of features. Because of this, feature flags enable many novel workflows that are incredibly useful to an [agile management](https://www.atlassian.com/agile/project-management) style and CI/CD environments.

1. **What is CI/CD pipeline?**

CI/CD stands for Continuous Integration/Continuous Delivery/Continuous Deployment. It is a set of practices that enable developers to integrate code changes more frequently, test those changes, and deliver software updates more quickly and with fewer errors.

A CI/CD pipeline consists of a series of automated processes that allow you to build, test, and deploy software more efficiently. These processes are typically executed by a CI/CD tool, which can be configured to run each time a change is made to the codebase.

There are typically several stages in a CI/CD pipeline, including:

1. Build: In this stage, the CI/CD tool compiles the source code and runs any necessary build tasks, such as installing dependencies or generating documentation.
2. Test: In this stage, the CI/CD tool runs automated tests to ensure that the code changes do not introduce any new bugs or regressions.
3. Package: In this stage, the CI/CD tool creates a deployable package or artifact, such as a Docker image or a ZIP file, that contains the compiled code and any necessary dependencies.
4. Deploy: In this stage, the CI/CD tool pushes the package to a staging or production environment, where it can be tested or released to users.
5. Monitor: In this stage, the CI/CD tool monitors the deployed application and sends alerts if any issues are detected.

By automating these processes, a CI/CD pipeline helps teams deliver software updates more quickly and with fewer errors, enabling them to respond to changing business needs more rapidly.

